



# Strawman Proposal

Terence Critchlow

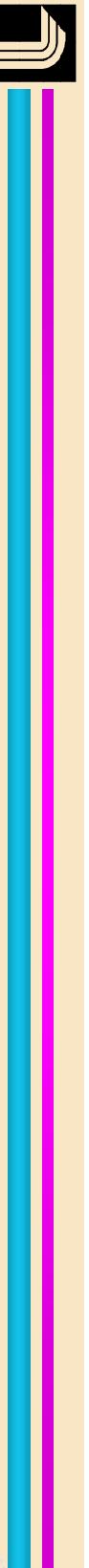
*Center for Applied Scientific Computing  
Lawrence Livermore National Laboratory  
[www.llnl.gov/CASC/people/critchlow](http://www.llnl.gov/CASC/people/critchlow)*

Dec 13, 2000

CASC

# The strawman design should be:

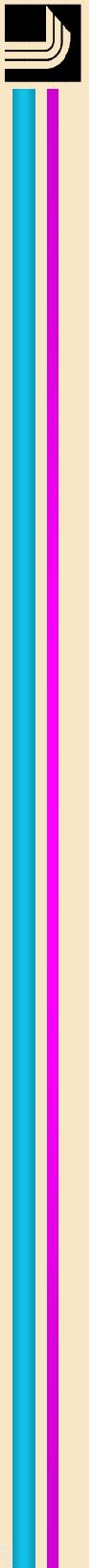
- A mechanism for describing genomics interfaces.
  - ❖ to create canonical interface definitions
  - ❖ to describe the legal interactions with a specific page.
- Flexible enough to utilize existing, domain-specific, semantic specifications.
- Simple enough to be used by a person.
- Powerful enough that a sufficiently smart wrapper generator could use it to automatically create a reasonable wrapper for the interface.



## Basic idea:

Define canonical interfaces and use their attributes to describe specific sources.

- Interface description
  - ❖ groups attributes into canonical interfaces
  - ❖ defines the properties of these attributes
- Source description
  - ❖ defines how a particular source uses these attributes
  - ❖ describes the expected results of a query



# Interface description (informal)

- Canonical interfaces:

- ❖ embody typical interactions with a variety of sources
- ❖ parameters, negative examples, typical input values

- Parameters:

- ❖ basic building block for interfaces and page descriptions
- ❖ name (unique), alternate names (tags), data type, textual semantic description

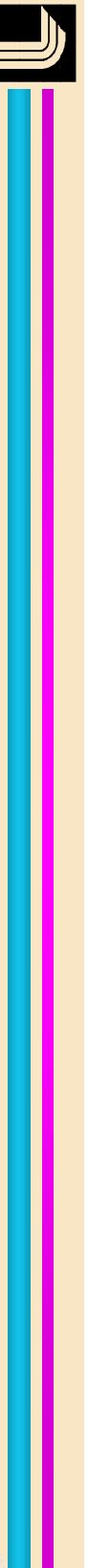
## Interface description (informal):

- Interface = name + required params + opt params + negative ex + constraints
- Parameter = name + datatype + description + tags + test values
- Constraint = operator + arguments

# Interface description example:

```
<interface>    <name> keyword search </name>
                <required> keyword </required>
                <not> sequence search </not> </interface>

<parameter>   <name> keyword </name>
                <datatype> <string /> </datatype>
                <tag> Search </tag>           <tag> Search for </tag>
                <test value> hemoglobin </test value>
                <test value> zinc finger </test value> </parameter>
```



# Source description (informal):

- Basic structural information:

- ❖ cgi script location, and associated thesaurus (interface description)
- ❖ results page location

- Result identification:

- ❖ define expected location of resulting attributes, methods for identifying them in a page
- ❖ handle complex results pages (i.e. indirection pages, delayed pages)

- Semantic interpretation:

- ❖ map between source attributes and reference thesaurus
- ❖ identify complex attribute transformations



## Source description (informal):

- Page desc = url + cgi script + thesaurus + query
- Query = input + output
- Input = input param + constraints
- Input param = script param + thes param(s) + confidence + transformation
- Output = base url + delay? + out param
- Out param = res params + thes param + confidence + transformations
- Param = name + datatype + param loc + required? + value

## Known weaknesses

Several tasks are currently hard to do, including:

- Combining indirection pages into a comprehensive view of the results.
- Representing constraints in a consistent and useable format.
- Incorporating semantics from existing ontologies.

This work was performed under the auspices of the U.S.  
Department of Energy by University of California Lawrence  
Livermore National Laboratory under contract No. W-7405-  
ENG-48.

CLSC